

**“Of course I’m committed to Agile.  
Just tell me what I’m going to get for my money and when I’m going to get it!”**

David Lambert  
Vice President, David Consulting Group

## 1 Objective

Agile at Scale causes stress and mistrust when the expectations of business leaders, finance managers, PMOs and technology teams are not aligned around, “what will be delivered, when?” for large Agile programs. In large organizations, this can manifest as seemingly endless risk management meetings as the business and the PMO fret about the uncertainty. Worse, some technology leaders seek to score political points by feeding off of the business angst to undermine their colleagues leading the implementation by accusing them of being “too Agile” or “not Agile enough.” Or both!

This article provides a high-level glimpse into how we can minimize the expectations gaps by borrowing some tools and methods from the past to provide initial Program Validation and Ongoing Monitoring.

## 2 Flavours of Agile

There are many flavours of Lean-Agile methodology out there. Many companies and individuals have livelihoods dependent on defining, coaching and training on the “one true way with all the answers.” We never claim to have all the answers so it’s worth clarifying the context for this article by stipulating the following:

- As Agile coaches and practitioners in large organization ourselves, we prefer some Agile methodologies over others but respect all. The approach described is independent of Agile methodology.
- We know that mature Agile organizations with mature programs and teams have solved some of the problems brought up here. Many have not and the credibility of Agile is at risk in the eyes of those who hold the purse strings.
- This paper is based on observed behaviour in at least six large organizations in the US and the UK.
- This approach is in the public domain and is used by many organizations – some of whom even use it to tackle the problem described in the first paragraph – but many Agilista’s regard it as unnecessary or simple too “traditional.”
- There are several tools from different providers that can be used, all of which have similar capabilities. We do not sell any of them.

### 3 A Typical Scenario

#### 3.1 The Challenge

To help us understand the problem, let's build an example that is a collage of several actual programs that we have worked on: Financial Services company, BigBank.

Big Banks is:

- Under pressure from its shareholders and clients to modernize its technology offering around its mortgage products
- Facing competition from several startups who are hoovering up business from millennials.
- Competing against quick, paperless and positive online client experiences that take the client from expression of interest on the website through information gathering to decision and closing.

Unfortunately, this presents the mortgage business heads at BigBank with a few problems because their market research has scored them low on "quick" and "paperless." The Operations Leaders at BigBank explain to the Business Leaders that going paperless will solve the speed problem AND significantly reduce the high costs of errors that they are struggling with in dealing with lots of paper forms that are complex and often contain errors. The business case practically writes itself so they book a meeting with their colleagues, the Tech. Leaders, to make sure that the paperless future, already named the BigBankLite Program, can be built for \$50M over three years.

#### 3.2 The Budget

Why \$50M over three years?

Three years is probably the longest time the Business Leaders can keep the CEO and shareholders at bay without a noticeable response to the market. \$50M is about half of the operational savings expected in the five years after implementation which gives a reasonable ROI and some buffer for the inevitable (in their eyes) Tech. overspend.

Preliminary discussions with Tech Leaders comparing the scope of the BigBankLite Program to previous programs together with quick calculations of numbers of Agile trains and teams and their costs over three years suggests that most of what needs to be done should be doable with \$50M. Unfortunately, it would later emerge that nobody could be found who thought to retain the napkin on which this initial estimate was written.

The BigBank Tech. Leaders collective mouths water at \$50M in incremental funding over three years so they are predisposed to agree to implement the program and worry about the scope issues and the weak estimates later. They patiently explain to their business colleagues that while they are sure that they can deliver for this amount in this timeframe, their Agile methodology and philosophy precludes the sort of commitment that the business is looking for – a simple "Yes" to the scope, budget and timeframe. The Tech. Leaders explain that while they used to make these commitments in the (good old) waterfall days, the Business Leaders must recall (they don't), that changes of scope close the deadline dates were a regular feature of those days as were a steady dribble of change requests throughout the program life. Hence, being up-front about the uncertainty is much better (isn't it?).

### 3.3 Big Banks Solution

The BigBankLite Program will be run as an Agile Program with two “trains”: one in-house and one at our Strategic Partner, ABC Inc. The former having most of the back-end expertise and the latter having most of the front-end expertise. The Tech. Leaders explain that even if they could commit to scope, budget and timeframe on an Agile Program internally, ABC would never commit on the external work unless all the requirements were detailed and agreed upfront - like in a waterfall project (“which we are not going to regress to!”).

BigBank’s Business Leaders and the newly appointed PMO team start to sweat. They know they will be held accountable for a successful outcome and green “traffic light” status reports throughout on budget consumed versus progress made. The problem is escalated, the Tech. Leaders ability to lead their organizations is called into question and they sheepishly agree that they can’t understand why they didn’t commit to the scope for \$50M in three years when they were first asked (we have been in several meeting where this has actually happened). Some mid-ranking but fearful Tech. staff present at this meeting (on condition of not speaking because they sometimes blurt out the truth) were later heard to justify this approach as being the lesser of two evils compared to explaining to the Business Leaders how Agile works – and, in any case, the Tech. Leaders had their fingers crossed behind their backs.

Fortunately, the BigBankLite Program’s Product Management and Product Owners have prepared a high-level list of the sort of features that they would like to see at the end of three years. It is a long list of bullet points with no details (but that’s ok – we’re Agile) and includes the business equivalents of teleportation, traversing black holes and so on. The discussions between business and technology start with an attempt to get the business to define the Minimum Viable Product (MVP) and when they need, in the three year program, they need the MVP delivered. The PMO team goes away to build some work breakdown structures which nobody will ever use but it soothes their nerves and keep them out of everyone’s hair.

Positive discussions with the introduction of some detail and some architectural requirements and constraints lead to agreement on a set of Features to be delivered at the end of year 1 as the MVP. Business and Tech. leaders go away happy, the PMO allocates the first years budget and set the status of the program as “Green.”

### 3.4 The Likely Outcome

BUT nobody has the faintest idea of the likelihood of being able to deliver the MVP at the end of year one with the available budget.

#### **Two Different Approaches to the Program Budget-Scope Projection under Agile at Scale**

We have seen two very different approaches at this point:

1. **Plow ahead or “It will be alright on the night.”** The first, and by far the most common, is to plow ahead. Within a month or two, the status of the Program is still “Green” (because we have lots of time so don’t worry about it) but the status of the MVP is “Yellow” because there is no way of knowing how realistic the deliverable is one year from now.

At this point, many Agile practitioners will assert that this is not an issue – we know the capacity of the train in story points per Program Increment and we know the T-shirts sizes of the MVP features so we can roughly estimate story points to be built. Divide the rough story points to be built by the capacity of the train and we have the duration in program increments. It couldn’t be easier.

For continuing enhancements to applications on an existing value stream, I would agree with these assertions and have a high comfort level. Unfortunately, the BigBankLite program and many other new programs like it are breaking new ground. New trains and even new teams are formed that are often from different value streams. Historical capacity and velocity number do not apply so new metrics have to be generated and we have to wait until the new teams/trains have stabilized – stormed, formed and normed – before we can make good forecasts. The T-shirt size estimates for the Features can be wildly out as the teams go through an intensive discovery period in the first, second and maybe even third PI's. While this approach plays out very appropriately from an Agile philosophy perspective, to the external business observer it looks like unpredictable, unmanaged chaos.

### 3.4.1 A better Approach and Outcome?

- 2. Program Estimation and Deliverable Monitoring.** We have also worked with companies who have adopted a disciplined and structured approach to estimating the scope of the program using long-established software estimation tools. The inputs to these tools are the size of the Features (analogous to the T-Shirt sizes in approach #1 but based on a more rigorous, repeatable sizing method) and a set of parameters reflecting the development context of the project (e.g. programming languages to be used, target architecture, team experience, etc.). The software estimation tools do not simply throw out an alternative number to help validate the Tech. leaders “expert” estimates, they use monte carlo analysis to identify the probabilities of a range of estimates.

We have found that companies that use this second approach are able to start a dialogue between the business and the Tech. leaders focused on a formal or informal sensitivity analysis around the output from the software estimation tool. It might revolve around removing Features from the MVP or extending the time period for delivery of the MVP to keep those Features in. It might involve looking at the impact if the initial assumptions about the team's competence were too optimistic or changing the onshore/offshore mix. With today's software estimation tools, all of these “what if” scenarios can be reviewed at the click of a couple of buttons by an expert user!

The key lesson from approach #2 is that high level, sophisticated estimates that are independent from the expert guesses of the Tech leaders and the newness of the Agile trains/teams and the newness of the problem being solved can be created at the start of the program that can be monitored, validated and adjusted throughout the program.

## 4 Conclusion

---

We mentioned that the tools need to be driven by an expert user. It is all too easy to tweak the input parameters of the tools to get the number you first thought of. An independent, objective individual or team driving the software tool is essential for the credibility of the results in the eyes of the business leaders AND the technology leaders. This is an important point psychologically as well as practically. Now the business leaders and technology leaders are looking at the parameters and probabilities together to get the best outputs and agree on an acceptable risk probability rather than the “traditional” haggling approach to getting a mutually acceptable estimate. Agile tells us to value conversations over contracts. It also helps that we are making data-driven decisions.